This listing of claims will replace all prior versions, and listings, of claims in the application:

The Status of the Claims

1. (Currently Amended) A method of computer operating system data management comprising ~~the steps of~~:

~~(a)~~ associating data management information with data input to a process; and

~~(b)~~ regulating operating system operations involving the data according to the data management information by:

disassembling an application to be executed to obtain machine code; and

modifying the obtained machine code of the application to include instructions for regulating the data according to the data management information.

2. (Original) The method of claim 1 wherein supervisor code administers the method by controlling the process at run time.

3. (Currently Amended) The method of claim 1, wherein, ~~the step (a)~~associating the data management information with the data input to the process comprises associating the data management information with the data as the data is read into a memory space.

4. (Currently Amended) The method of claim 1, wherein ~~the step (a)~~associating the data management information with the data input to the process comprises associating the data management information with at least one data sub-unit as the data is read into a memory space from a data unit comprising a plurality of data sub-units.

5. (Currently Amended) The method of claim 1, wherein ~~the step (a)~~associating the data management information with the data input to the process comprises associating the data management information with each independently addressable data unit that is read into the memory space.

6. (Original) The method of claim 2, wherein the data management information is written to a data management memory space under control of the supervisor code.

7. (Currently Amended) The method of claim 6 wherein the supervisor code comprises state machine automatons arranged to control the writing of the data management ~~info~~information to the data management memory space.

8. (Currently Amended) The method of claim 1, wherein ~~the step (b)~~regulating the operating system operation comprises: ~~sub-steps (b1)~~ identifying an operation involving the data; ~~(b2)~~ if the operation involves the data and is carried out within the process, maintaining an association between an output of the operation and the data management information; and ~~(b3)~~ if the operation involving the data includes a write operation to a location external to the process, selectively performing the operation dependent on the data management information.

9. (Currently Amended) The method of claim 8, wherein ~~the step (b1)~~identifying the operation comprises: ~~analysing~~analyzing process instructions to identify ~~operations~~the operation involving the data; and, providing instructions relating to the data management information with the ~~operations~~operation involving the data.

10. (Currently Amended) The method of claim 9, wherein the process instructions are ~~analysed~~analyzed as blocks, each block defined by operations up to a terminating condition.

11. (Currently Amended) The method of claim 1, in which code of an application is ~~analysed~~analyzed statically in order to create a control flow graph.

12. (Currently Amended) The method of claim 11, in which the code is ~~analysed~~analyzed before load time.

13. (Currently Amended) The method of claim 11, in which the code is ~~analysed~~analyzed at load time.

14. (Original) The method of claim 11, in which code of an application is instrumented to identify an entry point of a conditional structure in the code and an exit point of the conditional structure, and in which the entry points and exit points are identified from the control flow graph.

15. (Original) The method of claim 14, in which the conditional structure includes a conditional expression, a process has a tag associated with a program counter stack and when the entry point of a conditional structure is identified at run-time, a current tag is pushed further on the program counter stack, and a new tag associated with the conditional expression is added to the front of the counter stack.

16. (Original) The method of claim 15, in which when the exit point of a conditional structure is identified at run time, the tag from the entry point of the conditional structure is returned to the front of the counter stack.

17. (Currently Amended) The method of claim ~~14~~15, in which during all operations from an entry of the conditional structure, ~~the~~ tags of the locations in branching expressions are updated according to the tag of the program counter stack.

18. (Currently Amended) A computing platform including a processor for operating system data management, the computing platform comprising a data management unit, the data management unit arranged to associate data management information with data input to a process, and to regulate operating system operations involving the data according to the data management information by disassembling an application to be executed to obtain

machine code and modifying the obtained machine code of the application to include instructions for regulating the data according to the data management.

19. (Original) The computing platform of claim 18, further comprising a memory space, the computing platform arranged to load the process into the memory space and run the process under the control of the data management unit.

20. (Original) The computing platform of claim 18, wherein the data management information is associated with at least one data sub-unit as data is input to a process from a data unit comprising a plurality of sub-units.

21. (Original) The computing platform of claim 18, wherein the data management information is associated with each independently addressable data unit.

22. (Original) The computing platform of claim 18, wherein the data management unit comprises part of an operating system kernel space.

23. (Original) The computing platform of claim 22, wherein the operating system kernel space comprises a tagging driver arranged to control loading of a supervisor code into the memory space with the process.

24. (Original) The computing platform of claim 23, wherein the supervisor code controls the process at run time to administer the operating system data management unit.

25. (Currently Amended) The computing platform of claim 22, wherein the supervisor code is arranged to ~~analyse~~analyze instructions of the process to identify operations involving the data, and, ~~providing~~provide instructions relating to the data management information with the operations involving the data.

26. (Original) The computing platform of claim 23, wherein the memory space further comprises a data management information area under control of the supervisor code arranged to store the data management information.

27. (Original) The computing platform of claim 19, wherein the data management unit comprises a data filter arranged to identify data management information associated with data that is to be read into the memory space.

28. (Original) The computing platform of claim 27, wherein the data filter is arranged to associate data management information with data read into the memory space from predetermined sources, or alternatively is arranged to associate default data management information with data read into the memory space.

29. (Original) The computing platform of claim 18, wherein the data management unit further comprises a tag management module arranged to allow a user to specify data management information to be associated with data.

30. (Original) The computing platform of claim 18, wherein the data management unit comprises a tag propagation module arranged to maintain an association with the data that has been read into the process and the data management information associated therewith.

31. (Original) The computing platform of claim 30, wherein the tag propagation module is arranged to maintain an association between an output of operations carried out within the process and the data management information associated with the data involved in the operations.

32. (Original) The computing platform of claim 31, wherein the tag propagation module comprises state machine automatons arranged to maintain an association between an output of operations carried out within the process and the data management information associated with the data involved in the operations.

33. (Currently Amended) The computing platform of claim 18, in which code of an application is instrumented to identify an entry point of a conditional structure in the code and an exit point of the conditional structure, the computing platform further comprising a static code ~~analyser~~analyzer to identify conditional branch entry and exit points and a conditional tag propagator ~~for run-time propagation of~~to propagate, at runtime, tags associated with data storage locations included in the conditional structure.

34. (Currently Amended) An operating system data management method comprising ~~the step of~~: disassembling an application to be executed to obtain machine code; and

modifying the obtained machine code of the application to include instructions to ~~identifying~~identify data having data management information associated therewith when the data is to be read into a memory space.

35. (Currently Amended) The method of claim 34, further comprising ~~the step of~~: associating data management information with the data if the data is identified as having no data management information associated therewith.

36. (Original) The method of claim 34, wherein the data management information associated with data is read into the memory space with the data.

37. (Currently Amended) The method of claim 34, further comprising ~~the step of~~: maintaining an association between the data and the data management information when the

data is involved in operations within ~~the~~a process, and associating data management information with other data resulting from operations involving the data.

38. (Cancelled)

39. (Currently Amended) The method of claim 37, further comprising ~~the step of~~: examining the data management information when the data is to be involved in an operation external to the process, and allowing the operation if it is compatible with the data management information.

40. (Original) The method of claim 39, wherein the operation is blocked if it is not compatible with the data management information.

41. (Original) The method of claim 39, wherein the operation external to the process is compatible with the data management information subject to including the associated data management information with an output of the operation.

42. (Original) The method of claim 34, wherein the data management information identifies a set of permitted operations.

43. (Currently Amended) An operating system data management apparatus comprising:

a data management unit arranged to associate data management information with data input to a process, and to regulate operating system operations involving the data according to the data management information by disassembling an application to be executed to obtain machine code and modifying the obtained machine code of the application to include instructions to regulate the data according to the data management; and

~~a data filter arranged to~~ a processor to identify data having data management information associated therewith when that data is read into a memory space.

44. (Currently Amended) The apparatus of claim 43, wherein the ~~data filter comprises part of a data management unit, and~~processor is arranged to associate data management information with the data if the data is identified as having no data management information associated therewith.

45. (Currently Amended) The apparatus of claim 43, wherein ~~data management unit~~the processor is arranged to read the data management information associated with the data into the memory space with the data.

46. (Currently Amended) The apparatus of claim 43, ~~wherein the data comprises~~further comprising a tag propagation module arranged to maintain an association between the data and the data management information when the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

47. (Original) The apparatus of claim 46 wherein the tag propagation module comprises state machine automatons arranged to maintain an association between the data and the data management information when the data is involved in operations within the process, and to associate data management information with other data resulting from operations involving the data.

48. (Currently Amended) The apparatus of claim 46, wherein the tag propagation module is arranged to examine the data management information when the data is to be involved in an operation external to the process, and to cause the operation to be allowed if it is compatible with the data management information.

49. (Original) The apparatus of claim 48, wherein the tag propagation module is arranged to cause the operation to be blocked if the operation is not compatible with the data management information.

50. (Original) The apparatus of claim 48, wherein the tag propagation module is arranged to perform the operation external to the process subject to including the associated data management information with an output of the operation.

51. (Original) The apparatus of claim 43, wherein the data management information identifies a set of permitted operations.

52. (Currently Amended) A tangible computer readable medium storing a computer program including instructions configured to enable operating system data management in accordance with the method of operating system data management of claim 1.

53. (Currently Amended) A tangible computer readable medium storing a computer program including instructions configured to enable operating system data management in accordance with or the operating system data management method of claim 31.

54. (Currently Amended) A method of modifying computer code of an application, the method comprising ~~the steps of~~:

identifying conditional branches in ~~the~~machine code and instrumenting the machine code to provide information regarding ~~the~~ entry and exit points of the conditional structures; and

modifying the machine code to include instructions that, when executed, cause a computer to regulate the data according to the data management information.

55. (Original) The method of claim 54, in which the modification is carried out before load time.

56. (Original) The method of claim 54, in which the modification is carried out at load time.

57. (Currently Amended) The method of claims 54, further comprising ~~the step of~~ creating a control flow graph representation of the code and ~~analysing~~analyzing the conditional flow graph to identify conditional branches in the code.

58. (Currently Amended) An operating system stored on a tangible computer readable medium comprising an application code modifying unit arranged to perform the method of operating system data management of claim 1.

59. (Currently Amended) An operating system stored on a tangible computer readable medium comprising an application code modifying unit arranged to perform the operating system data management method of claim 34.